On the Expression Complexity of Equivalence and Isomorphism of Primitive Positive Formulas

Simone Bova[1], Hubie Chen[2], and Matthew Valeriote[3]

[1] Department of Mathematics
Vanderbilt University (Nashville, USA)
simone.bova@vanderbilt.edu
[2] Departament de Tecnologies de la Informació i les Comunicacions
Universitat Pompeu Fabra (Barcelona, Spain)
hubie.chen@upf.edu
[3] Department of Mathematics & Statistics
McMaster University (Hamilton, Canada)
matt@math.mcmaster.ca

Abstract. We study the complexity of equivalence and isomorphism on primitive positive formulas with respect to a given structure. We study these problems for various fixed structures; we present generic hardness and complexity class containment results, and give classification theorems for the case of two-element (boolean) structures.

1 Introduction

A *primitive positive* formula is a first-order formula defined from atomic formulas and equality of variables using conjunction and existential quantification. The class of primitive positive formulas includes–and is essentially equivalent to–the class of *conjunctive queries*, which is well-established in relational database theory as a pertinent and useful class of queries, and which has been studied complexity-theoretically from a number of perspectives (see for example [25, 21, 3]). In this paper, we study the complexity of two basic questions associated to primitive positive formulas:

(1) Given two such formulas having the same free variables and a relational structure, are the formulas equivalent over the structure? That is, do they admit the same satisfying assignments?
(2) Given two such formulas and a relational structure, are the formulas isomorphic over the structure? By isomorphic, we mean that there is a way to rename the free variables of one formula with the free variables of the other so that the formulas are equivalent.

We study both of these problems with respect to various fixed structures. That is, we parameterize each of these problems with respect to the structure to obtain a family of problems, containing one member for each structure, and study the resulting two families of problems. Following the terminology of Vardi [32], one may conceive of this as a study of the *expression complexity* of the two presented problems relative to various fixed structures. The suggestion here is that various relational structures–which may represent databases or knowledge bases, according to use–may possess structural characteristics that affect the complexity of the resulting problems, and this article initiates a study of this relationship between such structure and complexity.

Our investigation makes use of universal-algebraic tools and ideas that are of current interest in the study of the constraint satisfaction problem (CSP), which can be defined

Dagstuhl Seminar Proceedings 09441
The Constraint Satisfaction Problem: Complexity and Approximability
http://drops.dagstuhl.de/opus/volltexte/2010/2369

as the problem of deciding, given a primitive positive formula without free variables and a structure, whether or not the formula is true in the structure. As has been shown for CSPs, we demonstrate that to each structure one can associate an algebra in such a way that the complexity of all structures associated to the same algebra is the same, which permits the deployment of algebraic concepts in the study of our problems. We present a number of sufficient conditions for hardness for various complexity classes, as well as sufficient conditions for containment in various complexity classes. We then apply these results and further ideas to classify the complexity of these problems for boolean (two-element) structures.

Perspective and related work. Isomorphism problems have long been of interest in the theory of computational complexity (see for example [20, 27, 1, 30] and the references therein). The most prominent such problem, the graph isomorphism problem–decide if two given undirected graphs are isomorphic–is a famous example of a natural problem that is in NP but has defied classification as being in P or NP-complete. It is not NP-complete unless the polynomial hierarchy collapses [27], and not known to be in P.

Although Ladner's theorem [22] guarantees that there is a well-populated zone of intermediate problems that are in NP but not in P nor NP-complete (assuming that P does not equal NP), an interesting outcome of the study of constraint satisfaction problems and their variants [13] is that such problems tend to characterize established complexity classes; this can (and has been) taken as evidence for the thesis that natural problems can, by and large, be classified using existing complexity classes. A now classical example of this phenomenon is Schaefer's dichotomy theorem [26], which shows that every boolean CSP is either in P or is NP-complete; this result has been refined by a more recent result of Allender et al. [2] who showed that boolean CSPs are either complete for one of five established complexity classes (NP, P, NL, L, parity L) under AC^0 reductions, or of extremely low complexity (solvable in coNLOGTIME).

An additional motivation behind our investigation of the studied isomorphism problems was to see if any new degrees (interreducible sets of problems) of isomorphism problems would be revealed; or if these problems would simply inhabit established degrees of isomorphism problems, which might suggest that the natural degrees of isomorphism problems have already emerged. Of the five modes of behavior demonstrated by our isomorphism classification result, there is just one mode that appears somewhat new (see Theorem 10 (2)); the corresponding problems can be viewed as isomorphism on structures whose relations are definable by different types of sets of equations over the two-element field. Closely related problems have already been studied by Nordh [24]. For these problems, however, we show that their relationships to the major complexity classes in their vicinity, P and NP, are similar to those for graph isomorphism: these problems are in NP but not NP-complete unless the polynomial hierarchy collapses, and are not in P unless graph isomorphism is as well (graph isomorphism reduces to them).

We now turn to discuss some work directly related to our investigation, and which we draw upon. We develop some relationships between the complexity of the problems studied here and the complexity of the CSP, and in some concrete cases make use of the complexity results on the boolean CSP due to Allender et al. [2]. Böhler et al. [9, 8] have studied equivalence and isomorphism for boolean constraint satisfaction. The problems that they studied have a formulation similar to ours, with the key difference that the formulas that they studied are not permitted to contain existential quantification. The expressiveness of their formulas is thus in

general lower than that of ours, and this is reflected in the complexity results. Nordh [24] has conducted a study of equivalence and isomorphism of systems of equations over finite groups.

2 Problems Studied

In this section, we present the problems under study. Although we assume familiarity with the basics of first-order logic, we now review a few notions and present some conventions that we will use. For us, a *signature* is a finite set of relation symbols, each having an associated *arity*. A *relational structure* over a signature σ consists of a *domain* or *universe* B and, for each relation symbol $R \in \sigma$, a relation $R^{\mathbf{B}} \subseteq B^k$ where k is the arity of R. We assume that all relational structures under discussion have finite universe. A *primitive positive formula* (in short, *pp-formula*) on σ is a first-order formula formed using equalities on variables ($x = x'$), atomic formulas $R(x_1, \ldots, x_k)$ over σ, conjunction ($\wedge$), and existential quantification ($\exists$).

Definition 1. *The* primitive positive equivalence problem*,* PPEQ*, is the problem of deciding, given a relational structure* $\mathbf{B}$ *with signature* σ*, and two pp-formulas* ϕ*,* ϕ' *over* σ *having the same set of free variables* X*, whether* ϕ *and* ϕ' *are* equivalent *in* $\mathbf{B}$*, that is, whether for all* $f : X \to B$*,*

$$\mathbf{B}, f \models \phi \text{ if and only if } \mathbf{B}, f \models \phi'.$$

For every relational structure $\mathbf{B}$*, the problem* PPEQ($\mathbf{B}$) *is the primitive positive equivalence problem where the relational structure is fixed to be* $\mathbf{B}$*.*

Definition 2. *The* primitive positive isomorphism problem*,* PPISO*, is the problem of deciding, given a relational structure* $\mathbf{B}$ *with signature* σ*, and two pp-formulas* ϕ*,* ϕ' *over* σ *having the same set of free variables* X*, whether* ϕ *and* ϕ' *are* isomorphic *in* $\mathbf{B}$*, that is, whether there exists a bijection* $\pi : X \to X$ *such that for all* $f : X \to B$*,*

$$\mathbf{B}, f \models \phi \text{ if and only if } \mathbf{B}, f \circ \pi \models \phi',$$

where $f \circ \pi(\cdot) = f(\pi(\cdot))$*. For every relational structure* $\mathbf{B}$*, the problem* PPISO($\mathbf{B}$) *is the primitive positive isomorphism problem where the relational structure is fixed to be* $\mathbf{B}$*.*

We use BOOL-PPISO to denote the problem PPISO where $\mathbf{B}$ is required to have a universe of size 2.

The following facts are both known and straightforward to verify.

Proposition 1. PPEQ($\mathbf{B}$) *is in* Π_2^p *for every relational structure* $\mathbf{B}$*.*

Proposition 2. PPISO($\mathbf{B}$) *is in* Σ_3^p *for every relational structure* $\mathbf{B}$*.*

3 Algebra

We will study the problems of interest using techniques and notions from universal algebra. We follow the universal algebraic study of constraint satisfaction problems initiated in [12], and the results in this section are all either known or straightforward adaptations of results used to study constraint satisfaction. We begin by reviewing some definitions and concepts. An *algebra* is a pair $\mathbb{A} = (A, F)$ such that A is a nonempty set, called the *domain* or *universe* of the algebra, and F is a set of finitary operations on A. Let $\mathbb{A} = (A, F)$ be an algebra; a *term*

operation of $\mathbb{A}$ is a finitary operation obtained by composition of (1) operations in F and (2) projections on A, and a *polynomial operation* is a finitary operation obtained by composition of (1) operations in F, (2) projections on A and (3) constants from A. Two algebras are term (polynomially) equivalent if they have the same universe and the same term (polynomial) operations; we will generally be interested in algebras up to term equivalence. A clone is a set of operations that contains all projections and is closed under composition. Clearly, the set of all term operations of an algebra is the smallest clone containing the basic operations of the algebra. (See [28] for more details and discussion of these notions).

We say that an operation f is *idempotent* if the identity $f(x, \ldots, x) = x$ holds. An algebra is *idempotent* if all of its operations are idempotent; note that an idempotent algebra has only idempotent term operations. The *idempotent reduct* of an algebra $\mathbb{A} = (A, F)$, denoted by $I(\mathbb{A})$, is the algebra with universe A and whose operations are the idempotent term operations of $\mathbb{A}$.

Let B be a nonempty set, let f be an n-ary operation on B, and let R be a k-ary relation on B. We say that f *preserves* R (or f is a *polymorphism* of R, or R is *invariant* under f), if for every length n sequence of tuples $t_1, \ldots, t_n \in R$, denoting the tuple t_i by $(t_{i,1}, \ldots, t_{i,k})$, it holds that the tuple

$$f(t_1, \ldots, t_n) = (f(t_{1,1}, \ldots, t_{n,1}), \ldots, f(t_{1,k}, \ldots, t_{n,k}))$$

is in R. We extend this terminology to relational structures: an operation f is a polymorphism of a relational structure $\mathbf{B}$ if f is a polymorphism of every relation of $\mathbf{B}$. We use $\mathsf{Pol}(\mathbf{B})$ to denote the set of all polymorphisms of a relational structure $\mathbf{B}$, and use $\mathbb{A}_{\mathbf{B}}$ to denote the algebra $(B, \mathsf{Pol}(\mathbf{B}))$. For any relational structure $\mathbf{B}$, the set $\mathsf{Pol}(\mathbf{B})$ is a clone. Dually, for an operation f, we use $\mathsf{Inv}(f)$ to denote the set of all relations that are preserved by f, and for a set of operations F, we define $\mathsf{Inv}(F)$ as $\bigcap_{f \in F} \mathsf{Inv}(f)$. We will make use of the following result connecting the $\mathsf{Pol}(\cdot)$ and $\mathsf{Inv}(\cdot)$ operators to pp-definability.

Theorem 1. *(Geiger [14]/Bodcharnuk et al. [6]) Let* $\mathbf{B}$ *be a finite relational structure. The set of relations* $\mathsf{Inv}(\mathsf{Pol}(\mathbf{B}))$ *is equal to the set of relations that are pp-definable over* $\mathbf{B}$*.*

We will make use of the following fact, which is both well-known and straightforward to verify.

Proposition 3. *Let* $\mathbb{A}$ *be an algebra, and let* $R \subseteq A^k$ *be any relation. The smallest relation that contains* R *and is preserved by the operations of* $\mathbb{A}$ *is equal to*

$$\{f(t_1, \ldots, t_n) \mid n \geq 1,\ f \text{ is an } n\text{-ary term operation of } \mathbb{A}, \text{ and } t_1, \ldots, t_n \in R\}.$$

We associate to each algebra $\mathbb{A} = (A, F)$ the sets of problems

$$\mathsf{PPEQ}(\mathbb{A}) = \{\mathsf{PPEQ}(\mathbf{B}) \mid \mathbf{B} \text{ relational structure on } A \text{ with } F \subseteq \mathsf{Pol}(\mathbf{B})\},$$
$$\mathsf{PPISO}(\mathbb{A}) = \{\mathsf{PPISO}(\mathbf{B}) \mid \mathbf{B} \text{ relational structure on } A \text{ with } F \subseteq \mathsf{Pol}(\mathbf{B})\}.$$

For a complexity class $\mathcal{C}$, we say that the problem $\mathsf{PPEQ}(\mathbb{A})$ is in $\mathcal{C}$ if $\mathsf{PPEQ}(\mathbb{A}) \subseteq \mathcal{C}$. We say that the problem $\mathsf{PPEQ}(\mathbb{A})$ is $\mathcal{C}$-hard if $\mathsf{PPEQ}(\mathbb{A})$ contains a problem $\mathsf{PPEQ}(\mathbf{B})$ that is $\mathcal{C}$-hard. We say that the problem $\mathsf{PPEQ}(\mathbb{A})$ is $\mathcal{C}$-complete if it is both in $\mathcal{C}$ and $\mathcal{C}$-hard. We adopt a similar terminology for $\mathsf{PPISO}(\mathbb{A})$. The following result justifies these definitions.

Theorem 2. *Let* $\mathbf{B}$ *be a finite relational structure, and let* $\mathcal{C}$ *be a complexity class closed under logspace reduction.*

– $\mathsf{PPEQ}(\mathbf{B})$ *is in* $\mathcal{C}$ *if and only if* $\mathsf{PPEQ}(\mathbb{A}_{\mathbf{B}})$ *is in* $\mathcal{C}$.
– $\mathsf{PPEQ}(\mathbf{B})$ *is* $\mathcal{C}$*-hard if and only if* $\mathsf{PPEQ}(\mathbb{A}_{\mathbf{B}})$ *is* $\mathcal{C}$*-hard.*
– $\mathsf{PPEQ}(\mathbf{B})$ *is* $\mathcal{C}$*-complete if and only if* $\mathsf{PPEQ}(\mathbb{A}_{\mathbf{B}})$ *is* $\mathcal{C}$*-complete.*

And, the same results hold for $\mathsf{PPISO}(\cdot)$.

Throughout the paper, the notion of reduction used is logspace many-one reducibility, unless stated otherwise.

Proof. The third claim follows from the first two, so we turn to prove those. In the first claim, the ($\Leftarrow$) direction is obvious; in the second claim, the ($\Rightarrow$) direction is obvious. For the other directions, it suffices to show that each problem $\mathsf{PPEQ}(\mathbf{B}') \in \mathsf{PPEQ}(\mathbb{A}_{\mathbf{B}})$ reduces to $\mathsf{PPEQ}(\mathbf{B})$. By definition, we have $\mathsf{Pol}(\mathbf{B}) \subseteq \mathsf{Pol}(\mathbf{B}')$. It follows from the definition of $\mathsf{Inv}(\cdot)$ that $\mathsf{Inv}(\mathsf{Pol}(\mathbf{B})) \supseteq \mathsf{Inv}(\mathsf{Pol}(\mathbf{B}'))$. From Theorem 1 and the fact that each relation of $\mathbf{B}'$ is contained in the set $\mathsf{Inv}(\mathsf{Pol}(\mathbf{B}'))$, we have that each relation of $\mathbf{B}'$ is pp-definable over $\mathbf{B}$. The problem $\mathsf{PPEQ}(\mathbf{B}')$ can be reduced to $\mathsf{PPEQ}(\mathbf{B})$ by substituting each atomic formula in the original instance with a corresponding pp-definition.

The proofs are identical for $\mathsf{PPISO}(\cdot)$. □

Let $\mathbb{A} = (A, F)$ be an algebra. Let us say that a subset $B \subseteq A$ is preserved by the operations F of $\mathbb{A}$ if for every $n \geq 1$, every n-ary operation $f \in F$, and every $(b_1, \ldots, b_n) \in B$, it holds that $f(b_1, \ldots, b_n) \in B$. A *subalgebra* of $\mathbb{A}$ is an algebra of the form $(B, F|_B)$ where B is preserved by the operations of $\mathbb{A}$. Here, $F|_B$ denotes the set of all restrictions of operations in F to B, that is, $\{f|_B \mid f \in F\}$. A *congruence* of $\mathbb{A}$ is an equivalence relation $\theta \subseteq A \times A$ that is preserved by all operations of $\mathbb{A}$. When θ is a congruence of $\mathbb{A}$, the equivalence class of θ containing $a \in A$ is denoted by a^θ; and, for each operation $f \in F$, the operation f^θ defined by $f^\theta(a_1^\theta, \ldots, a_k^\theta) = (f(a_1, \ldots, a_k))^\theta$, is well-defined. We say that an algebra $\mathbb{A}_{\mathbf{B}}$ is a *homomorphic image* of $\mathbb{A}$ if it is isomorphic to the algebra (A^θ, F^θ), where $A^\theta = \{a^\theta \mid a \in A\}$ and $F^\theta = \{f^\theta \mid f \in F\}$.

Proposition 4. *Let* $\mathbb{B}$ *be a subalgebra or homomorphic image of an algebra* $\mathbb{A}$*. Then, for every problem* $\mathsf{PPEQ}(\mathbf{B}) \in \mathsf{PPEQ}(\mathbb{B})$*, there exists a problem* $\mathsf{PPEQ}(\mathbf{B}') \in \mathsf{PPEQ}(\mathbb{A})$ *such that* $\mathsf{PPEQ}(\mathbf{B})$ *logspace reduces to* $\mathsf{PPEQ}(\mathbf{B}')$*, and likewise for* $\mathsf{PPISO}(\cdot)$.

Proof. First, suppose that $\mathbb{B}$ is a subalgebra of $\mathbb{A}$. Consider a problem $\mathsf{PPEQ}(\mathbf{B}) \in \mathsf{PPEQ}(\mathbb{B})$; suppose that σ is the signature of $\mathbf{B}$. Define σ' to be a signature equal to σ but expanded by a relation symbol U of arity 1. Let $\mathbf{B}'$ be the relational structure over σ' with universe A where $R^{\mathbf{B}'} = R^{\mathbf{B}}$ for all $R \in \sigma$ and $U^{\mathbf{B}'} = B$. It follows from the definition of subalgebra that $\mathsf{PPEQ}(\mathbf{B}') \in \mathsf{PPEQ}(\mathbb{A})$. To reduce an instance ϕ, ϕ' of $\mathsf{PPEQ}(\mathbf{B})$ to $\mathsf{PPEQ}(\mathbf{B}')$, we simply restrict all variables to take on values in B, in each of the formulas. For each formula $\psi \in \{\phi, \phi'\}$, this can be done, for instance, by replacing every atomic formula $R(v_1, \ldots, v_k)$ by $R(v_1, \ldots, v_k) \wedge U(v_1) \wedge \ldots \wedge U(v_k)$ to obtain ψ', and then returning $\psi' \wedge U(x_1) \wedge \ldots \wedge U(x_n)$ where $\{x_1, \ldots, x_n\}$ are the free variables of ψ'.

Next, suppose that $\mathbb{B}$ is a homomorphic image of $\mathbb{A}$. We assume that the elements of $\mathbb{B}$ are $\{a^\theta \mid a \in A\}$ for a congruence θ of $\mathbb{A}$. Suppose that $\mathsf{PPEQ}(\mathbf{B}) \in \mathsf{PPEQ}(\mathbb{B})$, let σ denote the signature of $\mathbf{B}$, and define $\mathbf{B}'$ to be the structure over σ with universe A defined by $R^{\mathbf{B}'} = \{(a_1, \ldots, a_k) \mid (a_1^\theta, \ldots, a_k^\theta) \in R^{\mathbf{B}}\}$. Consider a pp-formula ψ. It is straightforward to verify that an assignment $(c_1, \ldots, c_n)$ satisfies ψ over $\mathbf{B}'$ if and only if the assignment $(c_1^\theta, \ldots, c_n^\theta)$ satisfies ψ over $\mathbf{B}$. Thus, a pair of pp-formulas ϕ, ϕ' is equivalent over $\mathbf{B}'$ if and only if it is equivalent over $\mathbf{B}$, and we have $\mathsf{PPEQ}(\mathbf{B}') = \mathsf{PPEQ}(\mathbf{B})$. □

The notion of a *variety* is typically defined on indexed algebras; a variety is a class of similar algebras that is closed under the formation of homomorphic images, subalgebras, and products. For our purposes here, however, we may note that the variety generated by an algebra $\mathbb{A}$, denoted by $\mathcal{V}(\mathbb{A})$, is known to be equal to $HSP(\{\mathbb{A}\})$, where the operator H (for instance) is the set of algebras derivable by taking homomorphic images of algebras in the given argument set. The power of an algebra (A, F) with respect to index set I is the algebra with universe $\Pi_{i\in I}A_i$, where each A_i is a copy of A, and which has an operation f' for each operation $f \in F$ that is defined to act as f on all coordinates.

We will make use of the language of tame congruence theory [17] to present some of our results. This theory associates a *typeset* to an algebra, which contains one or more of five *types*: (1) the unary type, (2) the affine type, (3) the boolean type, (4) the lattice type, and (5) the semilattice type. By extension, a typeset is associated to each variety, namely, the union of all typesets of algebras contained in a variety. A variety is said to *admit* a type if the type is contained in its typeset, and is otherwise said to *omit* the type.

We will make use of the following lemma, which generalizes a result found in [11]. A *factor* of $\mathbb{A}$ is a homomorphic image of a subalgebra of $\mathbb{A}$. An algebra is *strictly simple* if it is simple (has no non-trivial congruences) and has no non-trivial subalgebras.

Lemma 1. *([31]) Let $\mathbb{A}$ be a finite, idempotent algebra such that $\mathcal{V}(\mathbb{A})$ admits type i. Then, the algebra $\mathbb{A}$ has as a factor a strictly simple algebra of type at most i with respect to the ordering $1 < 2 < 3 > 4 > 5 > 1$.*

Szendrei classified all idempotent strictly simple algebras [29]; here, as in [23], we will make use of the following cases. By a *two-element set*, we mean a two-element algebra with no basic operations. An *affine algebra* is an algebra having an abelian group structure on its base set such that (1) $m(x, y, z) = x - y + z$ is a term of the algebra, and (2) every term of the algebra commutes with m. A *two-element semilattice* is an algebra isomorphic to the algebra $(\{0, 1\}, \{\wedge\})$. A *two-element lattice* is an algebra isomorphic to the algebra $(\{0, 1\}, \{\wedge, \vee\})$.

Lemma 2. *(follows from Szendrei [29]) Let $\mathbb{A}$ be a strictly simple idempotent algebra.*

- *If $\mathbb{A}$ has the unary type, then it is term equivalent to a two-element set.*
- *If $\mathbb{A}$ has the affine type, then it is an affine algebra.*
- *If $\mathbb{A}$ has the semilattice type, then it is term equivalent to a two-element semilattice.*
- *If $\mathbb{A}$ has the lattice type, then it is polynomially equivalent to a two-element lattice.*

4 Hardness Results

In this section, we present hardness results for the problems under study. We will make use of the following problems to give hardness results for the primitive positive isomorphism problem. The *graph isomorphism problem*, GI, is the problem of deciding whether two graphs $G = (V, E)$ and $G' = (V, E')$ are isomorphic, that is, whether there exists a bijection $\pi : V \to V$ such that (x, y) is in E if and only if $(\pi(x), \pi(y))$ is in E'. The problem GI is in NP, not known to be in P, and not NP-hard unless the polynomial hierarchy collapses [20]; it is also known to be hard for NL and other nondeterministic logarithmic space classes [30]. The *circuit isomorphism problem*, CI, is the problem of deciding whether two boolean circuits over the same input gates are isomorphic, that is, whether there exists a permutation of the input gates making both circuits compute the same boolean function. The problem CI is coNP-hard, in Σ_2^p, and

not Σ_2^p-hard unless the polynomial hierarchy collapses [1]. At the risk of feeling slightly guilty, we will overload our notation and also use GI, CI, and $\mathsf{BOOL\text{-}PPISO}$ to denote the class of problems that reduce to the problem at hand. For instance, GI will be used to denote the class of problems that reduce to GI; we will then be able to speak of, for instance, GI-hardness.

Our first hardness result shows, roughly speaking, that for a given algebra, the two problems studied here are harder than the constraint satisfaction problem over the idempotent reduct of the algebra. This result allows us to infer hardness results on these two problems from hardness results on the constraint satisfaction problem. In order to state the result, we introduce the following notions. Let $\mathbf{B}$ be a relational structure over signature σ. We use $\mathsf{CSP}(\mathbf{B})$ to denote the problem of deciding, given a pp-formula with no free variables over σ, whether or not the formula is true over $\mathbf{B}$. For an algebra $\mathbb{A}$, we define $\mathsf{CSP}(\mathbb{A})$ analogously to our definitions of $\mathsf{PPEQ}(\mathbb{A})$ and $\mathsf{PPISO}(\mathbb{A})$:

$$\mathsf{CSP}(\mathbb{A}) = \{\mathsf{CSP}(\mathbf{B}) \mid \mathbf{B} \text{ relational structure on } A \text{ with } F \subseteq \mathsf{Pol}(\mathbf{B})\}.$$

Theorem 3. *Let $\mathbb{A}$ be an algebra. Each problem $\mathsf{CSP}(\mathbf{B})$ in $\mathsf{CSP}(I(\mathbb{A}))$ reduces to both a problem in $\mathsf{PPEQ}(\mathbb{A})$ and a problem in $\mathsf{PPISO}(\mathbb{A})$.*

Proof. Let ϕ be an instance of $\mathsf{CSP}(\mathbf{B}) \in \mathsf{CSP}(I(\mathbb{A}))$. Let $\{b_1, \ldots, b_k\}$ denote the universe of B and suppose that σ is the signature of $\mathbf{B}$. Let σ' be the signature containing a relation symbol R' of arity $n+k$ for each relation symbol R of arity n in σ. We define a relational structure $\mathbf{B}'$ as follows. For each $R' \in \sigma'$, define $R'^{\mathbf{B}'}$ to be the smallest relation closed under the operations of $\mathbb{A}$ containing $Z = \{(b_1, \ldots, b_k, a_1, \ldots, a_n) \mid (a_1, \ldots, a_n) \in R^{\mathbf{B}}\}$. By Proposition 3, this smallest relation is equal to $\{f(t_1, \ldots, t_m) \mid m \geq 1, t_1, \ldots, t_m \in Z, f \text{ } m\text{-ary term operation of } \mathbb{A}\}$.

Define $\phi'(x_1, \ldots, x_k)$ to be the pp-formula derived from ϕ by replacing each atomic formula $R(v_1, \ldots, v_n)$ with $R'(x_1, \ldots, x_k, v_1, \ldots, v_n)$. (We assume that the variables $x_1, \ldots, x_k$ are new variables that do not occur in ϕ.) Define $\psi'(x_1, \ldots, x_k)$ to be the pp-formula

$$\exists v_1 \ldots \exists v_n (R'(x_1, \ldots, x_k, v_1, \ldots, v_n))$$

for any relation symbol $R' \in \sigma'$. We will interpret ψ' on $\mathbf{B}'$; notice that it does not matter which relation symbol $R' \in \sigma'$ we pick to define ψ', as all of the relations $R'^{\mathbf{B}'}$ with $R' \in \sigma'$, when projected onto the first k coordinates, are equal to the smallest relation containing $(b_1, \ldots, b_k)$ closed under the operations of $\mathbb{A}$; let us denote this common projection by $S \subseteq B^k$.

We claim that if ϕ is true, then ϕ' and ψ' are equivalent on $\mathbf{B}'$, and are not isomorphic on $\mathbf{B}'$ otherwise.

Suppose that ϕ is true. Then $(b_1, \ldots, b_k)$ clearly satisfies ϕ'. As ϕ' is a pp-formula, the relation T it defines is preserved by all polymorphisms of $\mathbf{B}'$ (by Theorem 1), and hence by all operations of $\mathbb{A}$. It follows that S is a subset of T; on the other hand, T is, by our definition of ϕ', a subset of S, and we have $S = T$. Hence ϕ' and ψ' are equivalent on $\mathbf{B}'$.

Suppose that ϕ is false. Then $(b_1, \ldots, b_k)$ does not satisfy ϕ'. This follows from the following claim: for each tuple of the form $(b_1, \ldots, b_k, c_1, \ldots, c_n) \in R'^{\mathbf{B}'}$, it holds that $(c_1, \ldots, c_n) \in R^{\mathbf{B}}$. This claim holds by the following reasoning: for such a tuple $(b_1, \ldots, b_k, c_1, \ldots, c_n) \in R'^{\mathbf{B}'}$, we have that there are m tuples $t_1, \ldots, t_m \in \{(b_1, \ldots, b_k, a_1, \ldots, a_n) \mid (a_1, \ldots, a_n) \in R^{\mathbf{B}}\}$, and a term operation f of $\mathbb{A}$ such that $f(t_1, \ldots, t_m) = (b_1, \ldots, b_k, c_1, \ldots, c_n)$; an operation f such that this holds is idempotent; and, the relation $R^{\mathbf{B}}$ is thus preserved by f by assumption, implying that $(c_1, \ldots, c_n) \in R^{\mathbf{B}}$. As mentioned in the previous paragraph, the relation T defined by ϕ' is a subset of S; since $(b_1, \ldots, b_k)$ is not in T, it follows that T is a proper subset of S, and that ϕ' and ψ' are not isomorphic. □

The next three theorems give hardness results on the studied problems based on the presence of types, in the sense of tame congruence theory. In particular, we give hardness results based on admitting the unary, semilattice, and lattice types.

Theorem 4. *Let $\mathbb{A}$ be a finite idempotent algebra such that $\mathcal{V}(\mathbb{A})$ admits the unary type. Then,* $\mathsf{PPEQ}(\mathbb{A})$ *is Π_2^p-hard, and* $\mathsf{PPISO}(\mathbb{A})$ *is* $\mathsf{BOOL\text{-}PPISO}$*-hard.*

Proof. By Lemma 1, there exists a strictly simple idempotent factor of $\mathbb{A}$ having type 1; by Lemma 2, this factor is term equivalent to the algebra $\mathbb{A}' = (\{0,1\}, \emptyset)$; by Proposition 4, it suffices to show that $\mathsf{PPEQ}(\mathbb{A}')$ is Π_2^p-hard, and that $\mathsf{PPISO}(\mathbb{A}')$ is $\mathsf{BOOL\text{-}PPISO}$-hard.

Let $\mathbf{B}$ be the relational structure over the domain $\{0,1\}$ and the signature $\sigma = \{R_0, R_1, R_2, R_3\}$ such that $R_0^{\mathbf{B}} = \{(x,y,z) \in \{0,1\}^3 \mid x \vee y \vee z\}$, $R_1^{\mathbf{B}} = \{(x,y,z) \in \{0,1\}^3 \mid \neg x \vee y \vee z\}$, $R_2^{\mathbf{B}} = \{(x,y,z) \in \{0,1\}^3 \mid \neg x \vee \neg y \vee z\}$, $R_3^{\mathbf{B}} = \{(x,y,z) \in \{0,1\}^3 \mid \neg x \vee \neg y \vee \neg z\}$. Clearly, $\mathsf{PPEQ}(\mathbf{B}) \in \mathsf{PPEQ}(\mathbb{A}')$ and $\mathsf{PPISO}(\mathbf{B}) \in \mathsf{PPISO}(\mathbb{A}')$. We prove that $\mathsf{PPEQ}(\mathbf{B})$ is Π_2^p-hard, and that $\mathsf{PPISO}(\mathbf{B})$ is $\mathsf{BOOL\text{-}PPISO}$-hard. The claimed hardness results follow by Proposition 4.

As regards $\mathsf{PPEQ}(\mathbf{B})$, the satisfiability problem of boolean formulas $\forall X \exists Y \phi$, where ϕ is a 3-CNF, and X and Y partition the variables of ϕ, is Π_2^p-hard. Letting ϕ' be the pp-formula over σ that corresponds to ϕ in the obvious way, it is immediate to verify that $\forall X \exists Y \phi$ is satisfiable if and only if $\exists Y \phi'$ is equivalent to the formula identically true over $\mathbf{B}$. As regards $\mathsf{PPISO}(\mathbf{B})$, let $\mathbf{B}', \phi, \phi'$ be an instance of $\mathsf{BOOL\text{-}PPISO}$. It is well-known that every boolean relation is pp-definable over $\mathbf{B}$, so by replacing atomic formulas with pp-definitions, from the formulas ϕ, ϕ' we may compute formulas $\overline{\phi}, \overline{\phi'}$ where $\mathbf{B}, f \models \phi$ if and only if $\mathbf{B}', f \models \overline{\phi}$, and likewise for ϕ' and $\overline{\phi'}$. Thus $(\phi, \phi') \in \mathsf{PPISO}(\mathbf{B}')$ if and only if $(\overline{\phi}, \overline{\phi'}) \in \mathsf{PPISO}(\mathbf{B}')$. □

We now present a lemma that facilitates the establishment of hardness results for the isomorphism problem. Let X be a set, let $\{X_1, \ldots, X_k\}$ be a partition of X, and let π be a permutation of X. We say that π *fixes* X_i if $\{\pi(x) \mid x \in X_i\} = X_i$.

Lemma 3. *Let σ be a signature, let $\mathbf{B}$ be a relational structure over σ, and let ϕ and ψ be pp-formulas on σ. For each $k \geq 2$, it is possible to construct in logspace, given a partition $\{X_1, \ldots, X_k\}$ of the set X of free variables of ϕ and ψ, two pp-formulas ϕ' and ψ' on σ satisfying: ϕ' and ψ' are isomorphic if and only if ϕ and ψ have an isomorphism that fixes $X_1, \ldots, X_k$.*

Proof. We prove the following: (1) If ϕ' and ψ' are isomorphic over $\mathbf{B}$, then ϕ' and ψ' have an isomorphism over $\mathbf{B}$ that fixes $X_1, \ldots, X_k$; (2) ϕ' and ψ' have an isomorphism over $\mathbf{B}$ that fixes $X_1, \ldots, X_k$ if and only if ϕ and ψ have an isomorphism over $\mathbf{B}$ that fixes $X_1, \ldots, X_k$. The proof follows.

Let $s_1 = 1$ and, for $2 \leq i \leq k$, let $s_i = |X_1|s_1 + \cdots + |X_{i-1}|s_{i-1}$. For every $i \in [k]$ and every $x \in X_i$, introduce s_i many new variables $\{x_j \mid j \in [s_i]\}$. Let $W_x = \{x\} \cup \{x_j \mid j \in [s_i]\}$, so that $|W_x| = 1 + s_i$. The pp-formula ϕ' on σ is constructed by conjoining to ϕ, for every $i \in [k]$, every $x \in X_i$, and every pair $y, y' \in W_x$, the equality constraint $y = y'$. The construction of ψ' is identical. Let X' be the free variables of ϕ' and ψ'.

Claim 1. Let π be an isomorphism of ϕ' and ψ', and let $\pi(x) = y$ and $\pi(x') = y'$. If $f(y) = f(y')$ for every assignment f to X' satisfying ϕ' over $\mathbf{B}$, then the permutation π', identical to π with the exception that $\pi'(x) = y'$ and $\pi'(x') = y$, is an isomorphism of ϕ' and ψ'.

Proof of Claim 1. Let f be an assignment to X' that models ϕ' over $\mathbf{B}$. Since by hypothesis $f(y) = f(y')$, we have that $f \circ \pi' = f \circ \pi$: indeed $f \circ \pi'(x) = f(\pi'(x)) = f(y') = f(y) = f(\pi(x)) = f \circ \pi(x)$, and $f \circ \pi'(x') = f(\pi'(x')) = f(y) = f(y') = f(\pi(x')) = f \circ \pi(x')$. Therefore, f models ϕ' if and only if $f \circ \pi$ models ψ' if and only if $f \circ \pi'$ models ψ'. □

For (1), let π be an isomorphism of ϕ' and ψ'. We show that there exists an isomorphism π' of ϕ' and ψ' that fixes X_i for every $i \in [k]$. Towards this aim, say that a permutation of X' is *correct* if, for every $i \in [k]$ and every $x \in X_i$, there exists $x' \in X_i$ such that $\pi(W_x) = \{\pi(y) \mid y \in W_x\} = W_{x'}$.

First consider X_k, and let $x \in X_k$. Note that by definition

$$|\pi(W_x)| = 1 + s_k > \left| \bigcup_{x' \in X \setminus X_k} W_{x'} \right|;$$

hence, there exists $x' \in X_k$ and $y \in W_x$ such that $\pi(y) = y' \in W_{x'}$. Let

$$\begin{aligned}
A &= \{\pi^{-1}(z) \mid z \in W_{x'}\} \setminus W_x = \{a_1, \ldots, a_n\},\\
B &= \{z \mid z \in W_{x'}\} \setminus \pi(W_x) = \{b_1, \ldots, b_n\},\\
C &= \{z \mid z \in W_x, \pi(z) \notin W_{x'}\} = \{c_1, \ldots, c_n\},\\
D &= \{\pi(z) \mid z \in W_x, \pi(z) \notin W_{x'}\} = \{d_1, \ldots, d_n\}.
\end{aligned}$$

Let $\pi_{k,x}$ be the permutation of X' defined as follows: $\pi_{k,x}$ is identical to π with the exception that, for $i = 1, \ldots, n$, $\pi_{k,x}(a_i) = \pi(c_i)$ and $\pi_{k,x}(c_i) = \pi(a_i)$. It is easy to verify that $\pi_{k,x}$ is an isomorphism of ϕ' and ψ'. Indeed, suppose for notation that $\pi(a_1) = b_1$ and $\pi(c_1) = d_1$, and consider the swap $\pi_{k,x}(a_1) = \pi(c_1) = d_1$ and $\pi_{k,x}(c_1) = \pi(a_1) = b_1$. Note that, for every assignment f satisfying ϕ', $f(b_1) = f(d_1)$. Indeed,

$$f(y') = f(b_1) = \cdots = f(b_n)$$

because the constraint $y' = b_1 = \cdots = b_n$ is in ϕ', and

$$f(\pi(y)) = f(\pi(c_1)) = \cdots = f(\pi(c_n))$$

because the constraint $y = c_1 = \cdots = c_n$ is in ψ'; hence, since $\pi(y) = y'$,

$$f(b_1) = f(y') = f(\pi(y)) = f(\pi(c_1)) = f(d_1).$$

But then, since π reaches $\pi_{k,x}$ by a sequence of swaps, and each swap is sound by Claim 1, $\pi_{k,x}$ is an isomorphism of ϕ' and ψ'.

Iterating over all $x \in X_k$, we obtain an isomorphism π_k of ϕ' and ψ' such that, for all $x \in X_k$, there exists $x' \in X_k$ such that $\pi_k(W_x) = W_{x'}$.

Possibly $\pi_k(x) = y \in W_{x'} \setminus \{x'\}$ for some $x \in X_k$. In this case, for some $y' \in W_x \setminus \{x\}$, $\pi_k(y') = x' \in W_{x'}$. But again, it is easy to check that the permutation that is identical to π_k with the exception that it sends x to x' and y' to y is an isomorphism of ϕ' and ψ'; hence iterating, we get an isomorphism ρ_k of ϕ' and ψ' that fixes X_k.

Now iterate over X_i for $i = k-1, \ldots, 1$, until an isomorphism ρ_1 of ϕ' and ψ' is obtained such that ρ_1 fixes X_i for every $i \in [k]$. The claim is settled by letting $\pi' = \rho$.

For (2, $\Rightarrow$), let π' be an isomorphism of ϕ' and ψ' that fixes X_i for every $i \in [k]$. By construction, $\pi'|_X$ is an isomorphism of ϕ and ψ, and clearly $\pi'|_X$ fixes X_i for every $i \in [k]$.

For (2, $\Leftarrow$), let ρ be an isomorphism of ϕ and ψ that fixes X_i for every $i \in [k]$. Then any permutation ρ' extending ρ to X', such that for every $i \in [k]$ and every $x \in X_i$, $\rho'(W_x \setminus \{x\}) = W_{x'} \setminus \{x'\}$ if and only if $\rho(x) = x'$, is an isomorphism of ϕ' and ψ'. □

Theorem 5. *Let $\mathbb{A}$ be a finite idempotent algebra such that $\mathcal{V}(\mathbb{A})$ admits the semilattice type. Then,* $\mathsf{PPEQ}(\mathbb{A})$ *is coNP-hard, and* $\mathsf{PPISO}(\mathbb{A})$ *is* CI*-hard.*

Proof. By Lemma 1, there exists a strictly simple idempotent factor of $\mathbb{A}$ having type at most 5. If this factor has type 1, both claimed hardness results follow from Theorem 4; otherwise, it has type 5, and is term equivalent to the algebra $\mathbb{A}' = (\{0,1\}, \{\wedge\})$ by Lemma 2. By Proposition 4, it suffices to show that $\mathsf{PPEQ}(\mathbb{A}')$ is coNP-hard, and that $\mathsf{PPISO}(\mathbb{A}')$ is CI-hard.

Let $\mathbf{B}$ be the relational structure over the domain $\{0,1\}$ and the signature $\sigma = \{T, F, I, N_2, H\}$ where $T^{\mathbf{B}} = \{(1)\}$, $F^{\mathbf{B}} = \{(0)\}$, $I^{\mathbf{B}} = \{(x,y) \in \{0,1\}^2 \mid x \to y\}$, $N_2^{\mathbf{B}} = \{(x,y) \in \{0,1\}^2 \mid \neg x \vee \neg y\}$, and $H^{\mathbf{B}} = \{(x,y,z) \in \{0,1\}^3 \mid x \wedge y \to z\}$. It is straightforward to verify that $\mathsf{PPEQ}(\mathbf{B}) \in \mathsf{PPEQ}(\mathbb{A}')$ and $\mathsf{PPISO}(\mathbf{B}) \in \mathsf{PPISO}(\mathbb{A}')$. We prove that $\mathsf{PPEQ}(\mathbf{B})$ is coNP-hard, and that $\mathsf{PPISO}(\mathbf{B})$ is CI-hard. The claimed hardness results follow by Proposition 4.

To this aim, we describe a logspace algorithm, $\phi(\cdot)$, that takes in input a boolean circuit C over the basis $\{\wedge, \neg\}$, say with input gates $x_1, \dots, x_n$, internal gates $x_{n+1}, \dots, x_{m-1}$, and output gate x_m, and returns in output a pp-formula $\phi(C)$ of σ, having free variables in

$$X = \{f_i, t_i \mid i \in [n]\},$$

and existentially quantified variables in

$$\{o_i \mid i \in [n]\} \cup \{f_i, t_i \mid i \in \{n+1, \dots, m\}\} \cup \{y_i \mid i \in [n+2]\}.$$

Note that, for $n \geq 3$, the pp-formula $H_{n+1}(x_1, \dots, x_n, x_{n+1})$ defined by

$$\exists y_1 \cdots \exists y_n \left(I(x_1, y_1) \wedge H(y_1, x_2, y_2) \wedge \cdots \wedge H(y_{n-1}, x_n, y_n) \wedge I(y_n, x_{n+1})\right)$$

pp-defines the relation $\{(x_1, \dots, x_n, x_{n+1}) \in \{0,1\}^{n+1} \mid (x_1 \wedge \cdots \wedge x_n) \to x_{n+1}\}$ over $\mathbf{B}$, and the pp-formula $N_n(x_1, \dots, x_n)$ defined by

$$H_{n+1}(x_1, \dots, x_n, x_{n+1}) \wedge F(x_{n+1})$$

pp-defines the relation $\{(x_1, \dots, x_n) \in \{0,1\}^n \mid \neg x_1 \vee \cdots \vee \neg x_n\}$ over $\mathbf{B}$.

The following is a logspace construction of $\phi(C)$: For every input gate x_i of C, $\phi(C)$ contains the *input* clauses

$$I(f_i, o_i),\ I(t_i, o_i),\ N_2(f_i, t_i);$$

for every internal gate x_i: if $x_i = \neg x_j$, $\phi(C)$ contains the *internal* clauses

$$I(f_j, t_i),\ I(t_j, f_i);$$

if $x_i = x_j \wedge x_k$, $\phi(C)$ contains the internal clauses

$$H(f_j, f_k, f_i),\ H(f_j, t_k, f_i),\ H(t_j, f_k, f_i),\ H(t_j, t_k, t_i);$$

eventually, corresponding to the output gate x_m, $\phi(C)$ contains the *output* clause

$$N_{n+1}(o_1, \dots, o_n, t_m).$$

Let $f' \colon X \to \{0,1\}$. Say that f' *corresponds* to $f \colon \{x_1, \dots, x_n\} \to \{0,1\}$ if $f(x_i) = 0$ implies $f'(f_i) = 1$, $f'(t_i) = 0$, and $f(x_i) = 1$ implies $f'(f_i) = 0$, $f'(t_i) = 1$.

Claim 1. f' does not model $\phi(C)$ over $\mathbf{B}$ if and only if either (1) $f'(f_i) = f'(t_i) = 1$ for some $i \in [n]$, or (2) f' corresponds to f and f satisfies C.

Proof of Claim 1. ($\Leftarrow$) If (1) holds, then some input clause $N_2(f_i, t_i)$ is false in $\mathbf{B}$ under f'. Suppose that (2) holds, and let f'' be an assignment to the existentially quantified variables of $\phi(C)$. The input clauses force f'' to send all the o_i's to 1. A routine inductive argument shows that the internal clauses force f'' to send t_m to 1 (respectively, f_m to 1) if the output gate x_m takes the value 1 (respectively, 0) under f; therefore, since f satisfies C, f'' sends t_m to 1. But then, the output clause $N_{n+1}(o_1, \ldots, o_n, t_m)$ is false in $\mathbf{B}$ under f'.

($\Rightarrow$) Suppose that neither (1) nor (2) hold. Suppose that f' corresponds to an assignment f not satisfying C. Let g be a partial assignment to the existentially quantified variables of $\phi(C)$ such that $g(o_i) = 1$ for all $i \in [n]$, and $g(f_i) = 1$ if and only if the value of gate x_i under the assignment f is 0 if and only if $g(t_i) = 0$ for all $i \in \{n+1, \ldots, m\}$. Note that in particular, $g(f_m) = 1$ and $g(t_m) = 0$. It is easy to check that g extends to a complete assignment f'' to the existentially quantified variables of $\phi(C)$ such that f' and f'' satisfy all the clauses of $\phi(C)$ in $\mathbf{B}$. Now suppose that $i \in [n]$ is such that $f'(f_i) = f'(t_i) = 0$. Note that, by hypothesis, there is no $i \in [n]$ such that $f'(f_i) = f'(t_i) = 1$. Then it is possible to extend a partial assignment g such that $g(o_i) = 0$ to a complete assignment f'' such that f' and f'' satisfy all the clauses of $\phi(C)$ in $\mathbf{B}$. □

As regards $\mathsf{PPEQ}(\mathbf{B})$, we reduce from the coNP-hard problem of deciding whether two boolean circuits over the same input gates are equivalent, that is, compute the same boolean function. Let C and C' be boolean circuits over $\{\wedge, \neg\}$ with the same input gates, and let $\phi(C)$ and $\phi(C')$ be the corresponding pp-formulas of σ. Exploiting Claim 1, it is easy to verify that C and C' are equivalent if and only if $\phi(C)$ and $\phi(C')$ are equivalent over $\mathbf{B}$.

As regards $\mathsf{PPISO}(\mathbf{B})$, we reduce from the circuit isomorphism problem. Let C and C' be boolean circuits over $\{\wedge, \neg\}$ with n many input gates $x_1, \ldots, x_n$, and let $\phi(C)$ and $\phi(C')$ be the pp-formulas of σ specified above, over the free variables $X = \{f_i, t_i \mid i \in [n]\}$. Let $\phi(C)'$ and $\phi(C')'$ be the pp-formulas of σ given by applying Lemma 3 to the blocks $F = \{f_i \mid i \in [n]\}$ and $T = \{t_i \mid i \in [n]\}$ of X, so that $\phi(C)'$ and $\phi(C')'$ are isomorphic if and only if $\phi(C)$ and $\phi(C')$ have an isomorphism that fixes F and T. We show that C and C' are isomorphic if and only if $\phi(C)'$ and $\phi(C')'$ are isomorphic over $\mathbf{B}$.

($\Rightarrow$) Let π be an isomorphism of C and C'. Define the permutation π' of X by putting $\pi'(f_i) = f_j$ and $\pi'(t_i) = t_j$ if and only if $\pi(x_i) = x_j$. Clearly, π' fixes F and T, and it is straightforward to verify that π' is an isomorphism of $\phi(C)$ and $\phi(C')$. Therefore, by Lemma 3, $\phi(C)'$ and $\phi(C')'$ are isomorphic.

($\Leftarrow$) Let π' be an isomorphism of $\phi(C)'$ and $\phi(C')'$. By Lemma 3, $\phi(C)$ and $\phi(C')$ have an isomorphism that fixes F and T.

Claim 2. Let $\pi \colon X \to X$ be an isomorphism of $\phi(C)$ and $\phi(C')$ that fixes F and T. For every $i \in [n]$, $\pi(f_i) = f_j$ and $\pi(t_i) = t_k$ implies $j = k$.

Proof of Claim 2. Let $f \colon X \to \{0, 1\}$ be identically 0 over $X \setminus \{f_j, t_k\}$, and such that $f(f_j) = f(t_k) = 1$. Then, $f \circ \pi(f_i) = f(\pi(f_i)) = f(f_j) = 1$ and $f \circ \pi(t_i) = f(\pi(t_i)) = f(t_k) = 1$. By Claim 1, $f \circ \pi$ does not model $\phi(C')$ over $\mathbf{B}$. Since π is an isomorphism of $\phi(C)$ and $\phi(C')$, $f \circ \pi \circ \pi^{-1} = f$ does not model $\phi(C)$ over $\mathbf{B}$. Noticing that f does not correspond to any assignment to the circuit C (take $n \geq 3$), by Claim 1 we have that $f(f_l) = f(t_l) = 1$ for some $l \in [n]$. But the only possibility is $j = k = l$. □

Define the permutation π of $\{x_1, \ldots, x_n\}$ by putting $\pi(x_i) = x_j$ if and only if $\pi'(f_i) = f_j$ and $\pi'(t_i) = t_j$. This definition is sound by Claim 2. It is straightforward to verify that π is an isomorphism of C and C'. □

Theorem 6. *Let $\mathbb{A}$ be a finite idempotent algebra such that $\mathcal{V}(\mathbb{A})$ admits the lattice type. Then,* $\mathsf{PPEQ}(\mathbb{A})$ *is NL-hard, and* $\mathsf{PPISO}(\mathbb{A})$ *is* GI*-hard.*

Proof. By Lemma 1, there exists a strictly simple idempotent factor of $\mathbb{A}$ having type at most 4. If this factor has type 1 or type 5, we invoke the proof of Theorem 4 and Theorem 5 to obtain the hardness result. In the case that we have such a factor of type 4, by Lemma 2, this factor is term equivalent to the algebra $\mathbb{A}' = (\{0,1\}, \{\wedge, \vee\})$.

Let $\mathbf{B}$ be the structure over signature $\{R, T, F\}$ with universe $\{0,1\}$ and where $R^{\mathbf{B}} = \{(0,0),(0,1),(1,1)\}$, $T^{\mathbf{B}} = \{(1)\}$ and $F^{\mathbf{B}} = \{(0)\}$. As the relation $R^{\mathbf{B}}$ is preserved by all polynomials of $\mathbb{A}'$, it is preserved by all terms of the factor; also, as the factor is idempotent, it preserves both $T^{\mathbf{B}}$ and $F^{\mathbf{B}}$. Hence, by Proposition 4, it suffices to show that $\mathbf{B}$ satisfies the described hardness results. The problem $\mathsf{CSP}(\mathbf{B})$ is known to be NL-hard [2], and we thus obtain that $\mathsf{PPEQ}(\mathbf{B})$ is NL-hard by the reduction that maps an instance ϕ of $\mathsf{CSP}(\mathbf{B})$ to the instance ϕ, ϕ' of $\mathsf{PPEQ}(\mathbf{B})$ where ϕ' is a true pp-formula with no free variables. It follows from [8, Lemma 14(1)] that $\mathsf{PPISO}(\mathbf{B})$ is GI-hard; their reduction can be implemented in logspace. □

5 Containment Results

This section presents complexity class containment results for the studied problems. We begin with an observation.

Proposition 5. *Let $\mathbf{B}$ be a relational structure. If* $\mathsf{CSP}(\mathbf{B})$ *is in* P*, then* $\mathsf{PPEQ}(\mathbf{B})$ *is in* coNP*, and* $\mathsf{PPISO}(\mathbf{B})$ *reduces to* CI*.*

The condition of having few subpowers was studied in [5]. Examples of algebras enjoying this property are those having a Maltsev term, that is, a ternary term m satisfying the identities

$$m(x,y,y) = m(y,y,x) = x$$

and those having a near-unanimity term, that is, a term f of arity greater than or equal to 3 satisfying the identities

$$f(y,x,x,\ldots,x) = f(x,y,x,\ldots,x) = \cdots = f(x,\ldots,x,y) = x.$$

We show that this condition in fact places the equivalence problem in P.

Theorem 7. *Let $\mathbf{B}$ be a relational structure. If $\mathbb{A}_{\mathbf{B}}$ has few subpowers, then* $\mathsf{PPEQ}(\mathbf{B})$ *is in* P*.*

Proof. Without loss of generality, we may assume that the structure $\mathbf{B}$ has all unary singletons as relations, since adding them will not change that $\mathbb{A}_{\mathbf{B}}$ has few subpowers. The following is a polynomial time algorithm for $\mathsf{PPEQ}(\mathbf{B})$. Given two formulas ϕ, ϕ' on free variables X, let $x_1, \ldots, x_n$ be an arbitrary ordering of the variables in X. Convert ϕ and ϕ' to prenex form, and let D, D' be extensions of the ordering $x_1, \ldots, x_n$ that include the existentially quantified

variables of ϕ and ϕ', respectively. Invoke the algorithm of [18] on ϕ and D to compute a succinct representation of the solution space (of the quantifier free part), and similarly for ϕ' and D'. Then, in the succinct representations, project away the existentially quantified variables to obtain succinct representations S and S' of the satisfying assignments of ϕ and ϕ' respectively. The algorithm then needs to check if S and S' are equal in the sense that they represent the same relation. As they are generating sets for the relations represented, it suffices to check that every tuple in each of S and S' is a solution of ϕ and ϕ'; this can be done by invoking the CSP decision procedure of [18]. For instance, if $(s_1, \ldots, s_n) \in S$ (and hence a solution of ϕ) then to determine if this tuple is a solution of ϕ', we need only check that the instance of $\mathsf{CSP}(\mathbf{B})$ obtained from ϕ' by substituting the variables x_i by the s_i has a solution. Since we assume that $\mathbf{B}$ has all unary singletons as relations, this instance does indeed belong to $\mathsf{CSP}(\mathbf{B})$. □

We now present results on structures having near-unanimity terms, giving different containment results depending on the complexity of the constraint satisfaction problem.

Theorem 8. *Let $\mathbf{B}$ be a relational structure. Suppose that $\mathbb{A}_\mathbf{B}$ has a near-unanimity term.*

- *If $\mathsf{CSP}(I(\mathbb{A}_\mathbf{B}))$ is in L, then the problem $\mathsf{PPEQ}(\mathbf{B})$ is in L, and the problem $\mathsf{PPISO}(\mathbf{B})$ reduces to GI under logspace reduction.*
- *If $\mathsf{CSP}(I(\mathbb{A}_\mathbf{B}))$ is in NL, then the problem $\mathsf{PPEQ}(\mathbf{B})$ is in NL.*
- *If $\mathsf{CSP}(I(\mathbb{A}_\mathbf{B}))$ is in P, then the problem $\mathsf{PPISO}(\mathbf{B})$ reduces to GI under polynomial time many-one reduction.*

Proof. Let $\mathbf{B}$ be a relational structure whose algebra $\mathbb{A}_\mathbf{B}$ has a near-unanimity term of arity k, and let ψ be a pp-formula with free variables X. It is known that, over $\mathbf{B}$, the formula ψ is satisfied by an assignment f if and only if for every subset $S \subseteq X$ of size less than or equal to $k-1$, the restriction $f|_S$ can be extended to a satisfying assignment of ψ [19].

Now consider the problem $\mathsf{PPEQ}(\mathbf{B})$. By expanding $\mathbf{B}$ if necessary, we may assume that $\mathbf{B}$ contains each of the constant relations $R_b^\mathbf{B} = \{(b)\}$ for $b \in B$; this is because adding these relations does not change the set of idempotent polymorphisms. To decide if ϕ, ϕ' are equivalent, it suffices to decide (by the initially stated fact) if, for every subset $S \subseteq X$ of size less than or equal to $k-1$, the sets $A_{S,\phi}$, $A_{S,\phi'}$ containing the assignments g on S that can be extended to satisfying assignments in ϕ and ϕ', are equal. This can be checked by looping over each such subset S and each assignment g on S; the body of the loop instantiates the assignment g in each formula by adding the constraint $R^b(v)$ when $g(v) = b$, and then calls a procedure for the CSP to check that one formula is true if and only if the other one is. When $\mathsf{CSP}(\mathbf{B})$ is in L, all of this can be carried out in logarithmic space. Similarly, when $\mathsf{CSP}(\mathbf{B})$ is in NL, this can be carried out in NL (note that since NL is closed under complementation, we may assume that the complement of $\mathsf{CSP}(\mathbf{B})$ is also in NL; we may then handle queries to the CSP by first guessing the outcome, and then querying either the CSP or its complement, continuing only if the guess was correct, and rejecting otherwise).

To reduce $\mathsf{PPISO}(\mathbf{B})$ to GI, for each of the formulas ϕ, ϕ' we may compute an equivalent quantifier-free pp-formula by looping over each subset S and assignment g on S (as above) and creating an atomic formula whose variables are S and where the relation contains the set of satisfying assignments. Although this relation may not be in $\mathbf{B}$, it is clearly pp-definable over $\mathbf{B}$ and hence is preserved by all polymorphisms of $\mathbf{B}$. The resulting quantifier-free pp-formulas may be reduced to GI by [9, Claim 22]. We can clearly carry this out in polynomial time if $\mathsf{CSP}(I(\mathbb{A}_\mathbf{B}))$ is in P. □

6 Classification of Boolean Case

In this section, we classify the complexity of the studied problems in the case of boolean structures. We begin with a lemma that will permit us to obtain hardness results for an algebra based on hardness results for the idempotent reduct of the algebra. Let us say that a clone over $\{0,1\}$ satisfies the *simple diagonal* property if for any operation f in the clone whose diagonal $f(x,\dots,x)$ is equal to a constant c, it holds that f itself is equal to c.

Lemma 4. *Let $\mathbb{A}$ be an algebra with universe $\{0,1\}$ whose clone of term operations satisfies the simple diagonal property. Then, for every problem $\mathsf{PPEQ}(\mathbf{B}) \in \mathsf{PPEQ}(I(\mathbb{A}))$, there exists a problem $\mathsf{PPEQ}(\mathbf{B'}) \in \mathsf{PPEQ}(\mathbb{A})$ such that $\mathsf{PPEQ}(\mathbf{B})$ reduces to $\mathsf{PPEQ}(\mathbf{B'})$, and likewise for $\mathsf{PPISO}(\cdot)$.*

Proof. Let $\mathbf{B}$ be any relational structure, let σ be its signature, and suppose that the relations of $\mathbf{B}$ are preserved by the operations of $I(\mathbb{A})$. Let $\mathbf{B'}$ be a relational structure, with the same universe and signature of $\mathbf{B}$, such that for every k-ary relation symbol R in σ, $R^{\mathbf{B'}}$ is the smallest $(k+2)$-ary relation that contains $\{(0,1,a_1,\dots,a_k) \mid (a_1,\dots,a_k) \in R^{\mathbf{B}}\}$ and is preserved by the operations of $\mathbb{A}$ (compare Proposition 3).

As regards the equivalence problem, let (ϕ,ψ) be an instance of $\mathsf{PPEQ}(\mathbf{B})$. Let X be the set of free variables of ϕ and ψ. We define an instance (ϕ',ψ') of $\mathsf{PPEQ}(\mathbf{B'})$, as follows. Let w_0 and w_1 be variables not occurring in ϕ or ψ. The pp-formula ϕ' of σ contains the constraint $R(w_0,w_1,x_1,\dots,x_k)$ if and only if ϕ contains the constraint $R(x_1,\dots,x_k)$. ψ' is defined similarly. We show that $(\phi,\psi) \in \mathsf{PPEQ}(\mathbf{B})$ if and only if $(\phi',\psi') \in \mathsf{PPEQ}(\mathbf{B'})$.

($\Leftarrow$) Let f be any assignment of X to $\{0,1\}$. We show that $\mathbf{B}, f \models \phi$ implies $\mathbf{B}, f \models \psi$ (the converse is similar). Suppose that f models ϕ over $\mathbf{B}$. Define $f'\colon X \cup \{w_0,w_1\} \to \{0,1\}$ such that $f'|_X = f$, $f'(w_0) = 0$, and $f'(w_1) = 1$. By construction, f' models ϕ' over $\mathbf{B'}$, thus by hypothesis, f' models ψ' over $\mathbf{B'}$; but by construction again, f models ψ over $\mathbf{B}$.

($\Rightarrow$) Let f be any assignment of $X \cup \{w_0,w_1\}$ to $\{0,1\}$. We show that $\mathbf{B'}, f \models \phi'$ implies $\mathbf{B'}, f \models \psi'$ (the converse is similar). Suppose that f models ϕ' over $\mathbf{B'}$. Let $R(w_0,w_1,x_1,\dots,x_k)$ be any constraint in ϕ'. The tuple $(f(w_0),f(w_1),f(x_1),\dots,f(x_k))$ is in $R^{\mathbf{B'}}$. By construction, there exist an m-ary term operation g of $\mathbb{A}$, and m many $(k+2)$-tuples $(0,1,a_{i,1},\dots,a_{i,k}) \in R^{\mathbf{B'}}$, $i \in [m]$, such that $g(0,0,\dots,0) = f(w_0)$, $g(1,1,\dots,1) = f(w_1)$, and $g(a_{1,j},a_{2,j},\dots,a_{m,j}) = f(x_j)$, $j \in [k]$. We distinguish three cases.

Case $f(w_0) = f(w_1) = a$: Since $\mathbb{A}$ has the simple diagonal property, by the above g is the constant a. Therefore $f(x_1) = \cdots = f(x_k) = a$. Without loss of generality, every variable in X occurs in ϕ', hence we conclude that f is identically a over X. But then, f models ψ' over $\mathbf{B'}$, because every relation of $\mathbf{B'}$ has $g = a$ as a polymorphism.

Case $f(w_0) = 0$ and $f(w_1) = 1$: By the above, g is idempotent. Then, g is in $I(\mathbb{A})$ and $(f(x_1),\dots,f(x_k)) \in R^{\mathbf{B}}$. Hence, $f|_X$ models ϕ over $\mathbf{B}$ and by hypothesis $f|_X$ models ψ over $\mathbf{B}$. By construction, f models ψ' over $\mathbf{B'}$.

Case $f(w_0) = 1$ and $f(w_1) = 0$: First note that the unary term operation that sends 0 to 1 and 1 to 0, in symbols $\neg x$, is a term operation of $\mathbb{A}$, because $\neg x = g(x,\dots,x)$. Hence the operation $\neg g$ is an idempotent term operation of $\mathbb{A}$, because $\neg g(x,\dots,x) = \neg(g(x,\dots,x)) = x$. Therefore, by construction, $(1,0,f(x_1),\dots,f(x_k))$ is in $R^{\mathbf{B'}}$ if and only if $(\neg f(x_1),\dots,\neg f(x_k))$ is in $R^{\mathbf{B}}$. But then, letting $\neg f|_X$ be the assignment of X to $\{0,1\}$ such that $\neg f|_X(x) = \neg f(x)$ for every $x \in X$, we have that $\neg f|_X$ models ϕ over $\mathbf{B}$. By hypothesis, $\neg f|_X$ models ψ over $\mathbf{B}$, and therefore, f models ψ' over $\mathbf{B}$.

As regards the isomorphism problem, let (ϕ, ψ) be an instance of $\mathsf{PPISO}(\mathbf{B})$. Let X be the set of free variables of ϕ and ψ. We define an instance (ϕ'', ψ'') of $\mathsf{PPISO}(\mathbf{B}')$ in two steps, as follows. ϕ' and ψ' are defined as for the equivalence problem. ϕ'' and ψ'' are defined by applying Lemma 3 to ϕ' and ψ' with the partition $\{w_0\}$, $\{w_1\}$, and X. We show that $(\phi, \psi) \in \mathsf{PPISO}(\mathbf{B})$ if and only if $(\phi'', \psi'') \in \mathsf{PPISO}(\mathbf{B}')$.

($\Leftarrow$) Let π'' be an isomorphism of ϕ'' and ψ'' over $\mathbf{B}'$. By Lemma 3, ϕ' and ψ' have an isomorphism π' that fixes $\{w_0\}$, $\{w_1\}$, and X. We show that $\pi = \pi'|_X$ is an isomorphism of ϕ and ψ over $\mathbf{B}$. Let $f\colon X \to \{0,1\}$. We show that $\mathbf{B}, f \models \phi$ implies $\mathbf{B}, f \circ \pi \models \psi$ (the converse is similar). Suppose that f models ϕ over $\mathbf{B}$. By construction, the assignment $f'\colon X \cup \{w_0, w_1\}$ such that $f'|_X = f$, $f'(w_0) = 0$, and $f'(w_1) = 1$, models ϕ' over $\mathbf{B}'$, then, $f' \circ \pi'$ models ψ' over $\mathbf{B}'$. Since $f' \circ \pi'(w_0) = f'(\pi'(w_0)) = f'(w_0) = 0$, $f' \circ \pi'(w_1) = f'(\pi'(w_1)) = f'(w_1) = 1$, and $f' \circ \pi'(x) = f'(\pi'(x)) = f'(\pi(x)) = f(\pi(x)) = f \circ \pi(x)$ for all $x \in X$, by construction we have that $f \circ \pi$ models ψ over $\mathbf{B}$.

($\Rightarrow$) Let π be an isomorphism of ϕ and ψ over $\mathbf{B}$. Let π' be the permutation of $X \cup \{w_0, w_1\}$ such that $\pi'|_X = \pi$, $\pi'(w_0) = w_0$, and $\pi'(w_1) = w_1$, that is, π' fixes $\{w_0\}$, $\{w_1\}$, and X. By the construction of ϕ' and ψ', π' is an isomorphism of ϕ' and ψ' over $\mathbf{B}'$. By Lemma 3, ϕ'' and ψ'' are isomorphic over $\mathbf{B}'$. □

Theorem 9 (Equivalence Classification). *Let* $\mathbf{B}$ *be a relational structure with universe* $\{0,1\}$*, and let* $\mathcal{V}$ *be the variety generated by the algebra* $I(\mathbb{A}_{\mathbf{B}})$*.*

1. *If* $\mathcal{V}$ *admits the unary type, then* $\mathsf{PPEQ}(\mathbf{B})$ *is* Π_2^p*-complete.*
2. *If* $\mathcal{V}$ *omits the unary type but admits the affine type, then* $\mathsf{PPEQ}(\mathbf{B})$ *is hard for parity L but contained in P.*
3. *If* $\mathcal{V}$ *omits the unary and affine types but admits the semilattice type, then* $\mathsf{PPEQ}(\mathbf{B})$ *is coNP-complete.*
4. *If* $\mathcal{V}$ *omits the unary, affine and semilattice types but admits the lattice type, then* $\mathsf{PPEQ}(\mathbf{B})$ *is NL-complete.*
5. *If* $\mathcal{V}$ *admits only the boolean type, then* $\mathsf{PPEQ}(\mathbf{B})$ *is in L.*

Proof. We consider each of the five cases in turn. In each case, the clone of term operations of $I(\mathbb{A}_{\mathbf{B}})$ is known from the analysis in [23]. From there, the possibilities for the clone of term operations of $\mathbb{A}_{\mathbf{B}}$ can be readily derived from Post's lattice [7]. By Theorem 2, it is sufficient to prove hardness results with respect to $\mathsf{PPEQ}(\mathbb{A}_{\mathbf{B}})$. We use the notation for clones introduced by [7].

Case 1: The algebra $I(\mathbb{A}_{\mathbf{B}})$ has term clone I_2. The term clone of $\mathbb{A}_{\mathbf{B}}$ is thus contained in N. As N satisfies the simple diagonal property, to obtain Π_2^p-hardness of $\mathsf{PPEQ}(\mathbf{B})$ it suffices by Lemma 4 to show that $\mathsf{PPEQ}(I(\mathbb{A}_{\mathbf{B}}))$ is Π_2^p-hard. This follows from Theorem 4. Containment of $\mathsf{PPEQ}(\mathbf{B})$ in Π_2^p follows from Proposition 1.

Case 2: The algebra $I(\mathbb{A}_{\mathbf{B}})$ has term clone L_2. As $x \oplus y \oplus z$ is contained in this clone, this operation is a term of $\mathbb{A}_{\mathbf{B}}$. This is a Maltsev operation and thus $\mathbb{A}_{\mathbf{B}}$ has few subpowers [5], and it follows from Theorem 7 that $\mathsf{PPEQ}(\mathbf{B})$ is in P. Parity L hardness of $\mathsf{PPEQ}(\mathbf{B})$ follows from Theorem 3 and the parity L hardness of $\mathsf{CSP}(I(\mathbb{A}_{\mathbf{B}}))$ [2].

Case 3: The algebra $I(\mathbb{A}_{\mathbf{B}})$ has term clone V_2 or E_2. The term clone of $\mathbb{A}_{\mathbf{B}}$ is thus contained in V or in E. As V and E satisfy the simple diagonal property, to obtain coNP-hardness of $\mathsf{PPEQ}(\mathbf{B})$ it suffices by Lemma 4 to show that $\mathsf{PPEQ}(I(\mathbb{A}_{\mathbf{B}}))$ is coNP-hard. This follows from Theorem 5. The inclusion of $\mathsf{PPEQ}(\mathbf{B})$ in coNP follows from Proposition 5 and the fact that $\mathsf{CSP}(\mathbb{A}_{\mathbf{B}})$ is in P [2].

Case 4: The algebra $I(\mathbb{A}_{\mathbf{B}})$ has term clone S^n_{00}, S^n_{10}, D_2, or M_2. For all of these term clones, $\mathsf{CSP}(I(\mathbb{A}_{\mathbf{B}}))$ is known to be NL-complete [2]. The NL-hardness of $\mathsf{PPEQ}(\mathbf{B})$ follows from Theorem 3 and the NL-hardness of $\mathsf{CSP}(I(\mathbb{A}_{\mathbf{B}}))$. Inclusion of $\mathsf{PPEQ}(\mathbf{B})$ in NL follows from Theorem 8.

Case 5: The algebra $I(\mathbb{A}_{\mathbf{B}})$ has term clone D_1, R_2, S^n_{02}, or S^n_{12}. Each of these term clones contains a near-unanimity operation and from [2] we have that $\mathsf{CSP}(I(\mathbb{A}_{\mathbf{B}}))$ is in L; thus $\mathsf{PPEQ}(\mathbf{B})$ is contained in L by Theorem 8. □

Theorem 10 (Isomorphism Classification). *Let* $\mathbf{B}$ *be a relational structure with universe* $\{0,1\}$*, and let* $\mathcal{V}$ *be the variety generated by the algebra* $I(\mathbb{A}_{\mathbf{B}})$*.*

1. *If* $\mathcal{V}$ *admits the unary type, then* $\mathsf{PPISO}(\mathbf{B})$ *is* $\mathsf{BOOL\text{-}PPISO}$*-complete.*
2. *If* $\mathcal{V}$ *omits the unary type but admits the affine type, then* $\mathsf{PPISO}(\mathbf{B})$ *is in NP and* GI*-hard, but not NP-complete unless the polynomial hierarchy collapses.*
3. *If* $\mathcal{V}$ *omits the unary and affine types but admits the semilattice type, then* $\mathsf{PPISO}(\mathbf{B})$ *is* CI*-complete.*
4. *If* $\mathcal{V}$ *omits the unary, affine and semilattice types but admits the lattice type, then* $\mathsf{PPISO}(\mathbf{B})$ *is* GI*-complete with respect to polynomial time many-one reduction.*
5. *If* $\mathcal{V}$ *admits only the boolean type, then* $\mathsf{PPISO}(\mathbf{B})$ *is either* GI*-complete or in* L*.*

To establish the isomorphism classification, we will make use of the following two lemmas. The first gives a hardness result for certain cases of the isomorphism problem.

Lemma 5. *Let* $\mathbf{B}$ *be a relational structure with universe* $\{0,1\}$ *such that the term clone of* $\mathbb{A}_{\mathbf{B}}$ *is contained in the clone* L *(which is generated by* $\oplus$ *and the constants* 0 *and* 1*). Then, the problem* $\mathsf{PPISO}(\mathbf{B})$ *is* GI*-hard.*

Proof. We make use of a construction and ideas from the paper by Nordh [24]. This paper shows the GI-hardness of deciding the isomorphism of systems of equations having the form $a \oplus b \oplus c = 1$. We will show that this isomorphism problem reduces to the problem of deciding the isomorphism of systems of equations having the form $a \oplus b \oplus c \oplus d = 0$; this suffices, since such an equation defines a 4-ary relation that is preserved by all operations in the clone L.

As in the paper of Nordh, we will make use of the concept of a *maximum* set of equations. We say that a set of equations U of a certain type (for instance, $a \oplus b \oplus c = 1$) is *maximum* if any equation of the same type that is entailed by U is included in U. It is straightforward to verify that for two maximum sets U_1, U_2 of equations (over the same type of equation), it holds that $U_1 \equiv U_2$, that is, U_1, U_2 have the same solutions, if and only if $U_1 = U_2$, by which we mean that the sets of equations are the same (up to commutativity of $\oplus$). As a consequence, two maximum systems of equations U_1, U_2 have isomorphic solution spaces if and only if there exists a permutation π such that $\pi(U_1) = U_2$.

Let T_1, T_2 be two sets of equations of the form $a \oplus b \oplus c = 1$ that are maximum. We show how to construct two sets of equations S_1, S_2 of the form $a \oplus b \oplus c \oplus d = 0$ that are isomorphic if and only if T_1, T_2 are isomorphic. For each i, define $S_i = \{(a \oplus b \oplus c \oplus w = 0) \mid (a \oplus b \oplus c = 1) \in T_i\}$, where w is a fresh variable not occurring in T_1 and T_2. By Lemma 3, we may create an instance of the problem $\mathsf{PPISO}(\mathbf{B})$ that is a yes instance if and only if S_1, S_2 are isomorphic via a permutation that fixes w. If S_1 and S_2 are isomorphic via such a permutation, then in particular they are isomorphic on assignments where $w = 1$, implying that T_1 and T_2 are isomorphic. On the other hand, if T_1 and T_2 are isomorphic via π, then by maximality they are in fact also isomorphic if each equation $a \oplus b \oplus c = 1$ is replaced by an equation $a \oplus b \oplus c = 0$. It follows that S_1 and S_2 are isomorphic via the extension of π that maps w to w. □

The next lemma gives a complexity upper bound for the isomorphism problem, assuming that the term clone has a particular clone.

Lemma 6. *Let* $\mathbf{B}$ *be a relational structure with universe* $\{0,1\}$ *such that the term clone of* $\mathbb{A}_{\mathbf{B}}$ *contains the operation* $x \oplus y \oplus z$*. Then, the problem* $\mathsf{PPISO}(\mathbf{B})$ *is not NP-complete unless the polynomial hierarchy collapses.*

Proof. It is known that relations preserved by the given operation are equivalent to the conjunction of linear equations over $\{0,1\}$. We may hence rewrite our formulas so that in place of atomic formulas, they contain such linear equations. By using Gaussian elimination, existentially quantified variables may be eliminated in a way that preserves the set of satisfying assignments. Hence, in polynomial time we may convert pp-formulas over $\mathbf{B}$ to systems of linear equations over $\{0,1\}$.

We give a constant-round interactive protocol for non-isomorphism of systems of linear equations over $\{0,1\}$, which is based on the protocol for graph non-isomorphism [15]. By the results of [16, 4, 10], this implies that the corresponding isomorphism question is not NP-complete, unless the polynomial hierarchy collapses. This resolves an open question of Nordh [24].

Given two systems of equations S_1, S_2 over the same variables X, the verifier does the following. He first checks satisfiability for each of the systems; if one or both are not satisfiable, he accepts or rejects accordingly. Next, he picks a random $b \in \{1,2\}$, and works with S_b. Throughout, we assume that equations have the form $a_1x_1 \oplus \cdots \oplus a_nx_n = c$ where $X = \{x_1, \ldots, x_n\}$ and $a_1, \ldots, a_n, c \in \{0,1\}$. Let us say that a variable v in a system of equations is obedient if, for any assignment g to the other variables, at most one of the extensions $g[v \to 0]$, $g[v \to 1]$ satisfies the system. Observe that a variable appears in an equation if and only if it is obedient. The verifier next picks a random variable v from those appearing in an equation, selects an equation E in which it occurs, and substitutes away all instances of the variable v in other equations based on E. The verifier then records (v, E), eliminates it from the system, and repeats, picking a random variable from the remaining equations.

When this process has terminated, any assignment to the non-recorded variables can be extended uniquely to a full assignment including the recorded variables. By using back substitution, every recorded variable can be written uniquely as a linear combination of non-recorded variables. The verifier creates an equation for each recorded variable based on this linear combination, randomly renames all of the variables, and outputs the result.

The verifier then sends the resulting system to prover. Prover's job is to attempt to identify which b was originally picked. As the transformation performed by verifier preserves the set of satisfying assignments up to permutation of variables, prover can do this if the original systems were not isomorphic. If the original systems were isomorphic, we claim that prover succeeds with probability $1/2$. This is because if the original systems were isomorphic, the distribution of formulas produced by the verifier for each choice of b is identical. In particular, suppose that π is an isomorphism from S_1 to S_2. The probability that the verifier chooses a sequence $v_1, \ldots, v_k$ of recorded variables when run on S_1 is equivalent to the probability that the verifier chooses the sequence $\pi(v_1), \ldots, \pi(v_k)$ of recorded variables when run on S_2. □

Proof. (Proof of Theorem 10) As in the proof of Theorem 9, we consider each of the five cases in turn. In each case, the clone of term operations of $I(\mathbb{A}_{\mathbf{B}})$ is known from the analysis in [23]. From there, the possibilities for the clone of term operations of $\mathbb{A}_{\mathbf{B}}$ can be readily derived

from Post's lattice [7]. By Theorem 2, it is sufficient to prove hardness results with respect to $\mathsf{PPISO}(\mathbb{A}_{\mathbf{B}})$. We use the notation for clones introduced by [7].

Case 1: The algebra $I(\mathbb{A}_{\mathbf{B}})$ has term clone I_2. The term clone of $\mathbb{A}_{\mathbf{B}}$ is thus contained in N. As all operations in N satisfy the simple diagonal property, to obtain BOOL-PPISO-hardness it suffices to show by Lemma 4 that $\mathsf{PPISO}(I(\mathbb{A}_{\mathbf{B}}))$ is BOOL-PPISO-hard. This follows from Theorem 4. Containment of $\mathsf{PPISO}(\mathbf{B})$ in BOOL-PPISO is clear.

Case 2: The algebra $I(\mathbb{A}_{\mathbf{B}})$ has term clone L_2. As $x \oplus y \oplus z$ is contained in this clone, this operation is a term of $\mathbb{A}_{\mathbf{B}}$. It follows from Lemma 6 that $\mathsf{PPISO}(\mathbf{B})$ is not NP-complete unless the polynomial hierarchy collapses. By Theorem 9, the problem $\mathsf{PPEQ}(\mathbf{B})$ is in P, from which it follows that $\mathsf{PPISO}(\mathbf{B})$ is in NP.

From Post's lattice, we have that the term clone of $\mathbb{A}_{\mathbf{B}}$ is contained in L. The GI-hardness of $\mathsf{PPISO}(\mathbf{B})$ follows from Lemma 5.

Case 3: The algebra $I(\mathbb{A}_{\mathbf{B}})$ has term clone V_2 or E_2. From Theorem 5, we obtain that $\mathsf{PPISO}(I(\mathbb{A}_{\mathbf{B}}))$ is CI-hard. Hence, by Lemma 4, $\mathsf{PPISO}(\mathbb{A}_{\mathbf{B}})$ is CI-hard, and by Theorem 2, we conclude that $\mathsf{PPISO}(\mathbf{B})$ is CI-hard. The inclusion of $\mathsf{PPISO}(\mathbf{B})$ in CI follows from Proposition 5 and the fact that $\mathsf{CSP}(\mathbb{A}_{\mathbf{B}})$ is in P [2].

Case 4: The algebra $I(\mathbb{A}_{\mathbf{B}})$ has term clone S^n_{00}, S^n_{10}, D_2, or M_2. In all of these term clones, there is a near-unanimity operation. We obtain that $\mathsf{PPISO}(\mathbf{B})$ polynomial time reduces to GI by Theorem 8. The GI-hardness of $\mathsf{PPISO}(\mathbf{B})$ follows from [8, Theorem 7(2)].

Case 5: The algebra $I(\mathbb{A}_{\mathbf{B}})$ has term clone D_1, R_2, S^n_{02}, or S^n_{12} (for some $n \geq 2$). Each of these term clones contains a near-unanimity operation and from [2] we have that $\mathsf{CSP}(\mathbf{B})$ is in L; thus, $\mathsf{PPISO}(\mathbf{B})$ logspace reduces to GI by Theorem 8.

In the case that the term clone of $I(\mathbb{A}_{\mathbf{B}})$ is of the form S^n_{02}, the term clone of $\mathbb{A}_{\mathbf{B}}$ is contained in S^2_0, and the relation $x \vee y$ is preserved by the term clone of $\mathbb{A}_{\mathbf{B}}$. The problem GI can be reduced to a structure $\mathbf{B}$ having the relation $R^{\mathbf{B}} = \{(x, y) \mid x \vee y\}$ by creating a formula for each graph containing an atomic formula $R(v, v')$ for each edge $\{v, v'\}$. The case of S^n_{12} is dual.

In the other two cases, where the term clone of $I(\mathbb{A}_{\mathbf{B}})$ is equal to either D_1 or R_2, all operations in D_1 are term operations of $\mathbb{A}_{\mathbf{B}}$. By the 2-decomposability [19] of the relations, we know that the set of satisfying assignments for a pp-formula with free variables X is equivalent to the conjunction of constraints $x \neq y$, x (x is true), and $\neg x$ (x is false); moreover, given one of these constraints, we may determine if it is entailed in logspace. Now consider two pp-formulas ϕ, ϕ'. In logspace, we can verify if both are satisfiable; if not, we can immediately determine isomorphism and either reject or accept. Now, consider the graphs G, G' where two variables x, x' are adjacent if there is an entailed constraint $x \neq x'$ in ϕ (respectively, ϕ'). Our logspace algorithm is as follows. For each of the formulas $\psi \in \{\phi, \phi'\}$, we loop over all variables v.

On variable v, we compute a triple t_v defined as (m, n, s) where m is the number of vertices reachable by an even number of G-steps from v, n is the number of vertices reachable by an odd number of G-steps from v, and s is equal to $a \in \{0, 1\}$ if v must be equal to a in any satisfying assignment, and equal to $*$ otherwise. We then count the number of connected components in G having a vertex w with $t_w = t_v$, and likewise for G'; if the numbers are different, we reject. (Looping over all connected components can be done by looping over all variables in order, and considering the component of a variable if and only if the variable is not connected to a variable coming before it in the order.)

If the algorithm loops over all variables in both formulas without rejecting, then the formulas are isomorphic, and the algorithm accepts. □

References

1. M. Agrawal and T. Thierauf. The Formula Isomorphism Problem. *SIAM Journal on Computing*, 30(3):990–1009, 2000.
2. E. Allender, M. Bauland, N. Immerman, H. Schnoor, and H. Vollmer. The Complexity of Satisfiability Problems: Refining Schaefer's Theorem. In *Proceedings of the 30th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, 2005.
3. A. Atserias. Conjunctive Query Evaluation by Search-Tree Revisited. *Theoretical Computer Science*, 371(3):155–168, 2007.
4. L. Babai and S. Moran. Arthur-Merlin Games: A Randomized Proof System, and a Hierarchy of Complexity Classes. *Journal of Computer and System Sciences*, 36(2):254 – 276, 1988.
5. J. Berman, P. Idziak, P. Markovic, R. McKenzie, M. Valeriote, and R. Willard. Varieties with Few Subalgebras of Powers. To appear in *Transactions of the American Mathematical Society*.
6. V. Bodnarchuk, L. Kaluzhnin, V. Kotov, and B. Romov. Galois Theory for Post Algebras. I, II. *Cybernetics*, 5:243–252, 531–539, 1969.
7. E. Böhler, N. Creignou, S. Reith, and H. Vollmer. Playing with Boolean Blocks, Part I: Post's Lattice with Applications to Complexity Theory. *ACM SIGACT-Newsletter*, 34(4):38–52, 2003.
8. E. Böhler, E. Hemaspaandra, S. Reith, and H. Vollmer. The Complexity of Boolean Constraint Isomorphism. *arXiv:cs/0306134v2*.
9. E. Böhler, E. Hemaspaandra, S. Reith, and H. Vollmer. Equivalence and Isomorphism for Boolean Constraint Satisfaction. In *Proceedings of the 16th International Workshop on Computer Science Logic (CSL)*, 2002.
10. R. Boppana, J. Hastad, and S. Zachos. Does co-NP have Short Interactive Proofs? *Information Processing Letters*, 25(2):127–132, 1987.
11. A. Bulatov and P. Jeavons. Algebraic Structures in Combinatorial Problems. Technical Report MATH-AL-4-2001, Technische Universitat Dresden, 2001.
12. A. Bulatov, P. Jeavons, and A. Krokhin. Classifying the Complexity of Constraints using Finite Algebras. *SIAM Journal on Computing*, 34(3):720–742, 2005.
13. N. Creignou, S. Khanna, and M. Sudan. *Complexity Classification of Boolean Constraint Satisfaction Problems*. SIAM Monographs on Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics, 2001.
14. D. Geiger. Closed Systems of Functions and Predicates. *Pacific Journal of Mathematics*, 27:95–100, 1968.
15. O. Goldreich, S. Micali, and A. Wigderson. Proofs that Yield Nothing but their Validity or all Languages in NP Have Zero-Knowledge Proof Systems. *Journal of the ACM*, 38(3):690–728, 1991.
16. S. Goldwasser and M. Sipser. Private Coins versus Public Coins in Interactive Proof Systems. In *Proceedings of the 18th Annual ACM Symposium on Theory of Computing (STOC)*, 1986.
17. D. Hobby and R. McKenzie. *The Structure of Finite Algebras*, volume 76 of *Contemporary Mathematics*. American Mathematical Society, 1988.
18. P. Idziak, P. Markovic, R. McKenzie, M. Valeriote, and R. Willard. Tractability and Learnability Arising from Algebras with Few Subpowers. In *Proceedings of the 22nd Annual IEEE Symposium on Logic in Computer Science (LICS)*, 2007.
19. P. Jeavons, D. Cohen, and M. Cooper. Constraints, Consistency, and Closure. *Articial Intelligence*, 101(1-2):251–265, 1998.
20. J. Kobler, U. Schöning, and J. Toran. *The Graph Isomorphism Problem: Its Structural Complexity*. Birkhäuser, 1993.
21. P. Kolaitis and M. Vardi. Conjunctive-Query Containment and Constraint Satisfaction. *Journal of Computer and System Sciences*, 61:302–332, 2000.
22. R. Ladner. On the Structure of Polynomial Time Reducibility. *Journal of the ACM*, 22(1):155–171, 1975.
23. B. Larose and P. Tesson. Universal Algebra and Hardness Results for Constraint Satisfaction Problems. To appear in *Theoretical Computer Science*.
24. G. Nordh. The Complexity of Equivalence and Isomorphism of Systems of Equations over Finite Groups. *Theoretical Computer Science*, 345(2-3):406–424, 2005.
25. C. Papadimitriou and M. Yannakakis. On the Complexity of Database Queries. *Journal of Computer and System Sciences*, 58(3):407–427, 1999.
26. T. Schaefer. The Complexity of Satisfiability Problems. In *Proceedings of the 10th Annual ACM Symposium on Theory of Computing (STOC)*, 1978.
27. U. Schoning. Graph Isomorphism is in the Low Hierarchy. *Journal of Computer and System Sciences*, 37(3):312–323, 1988.

28. A. Szendrei. *Clones in Universal Algebra*, volume 99 of *Seminaires de Mathematiques Superieures*. University of Montreal, 1986.
29. A. Szendrei. *A Survey on Strictly Simple Algebras and Minimal Varieties*, volume 19 of *Research and Exposition in Mathematics*, pages 209–239. Heldermann Verlag, 1992.
30. J. Toran. On the Hardness of Graph Isomorphism. *SIAM Journal on Computing*, 33(5):1093–1108, 2004.
31. M. Valeriote. A Subalgebra Intersection Property for Congruence Distributive Varieties. To appear in *Canadian Journal of Mathematics*.
32. M. Vardi. The Complexity of Relational Query Languages. In *Proceedings of the 18th Annual ACM Symposium on Theory of Computing (STOC)*, 1982.